複数オブジェクトの移動軌跡からの周期的なパターン検出

和田　康嗣† 　原田　圭† 　佐々木勇和† 　鬼塚　真†

† 大阪大学大学院情報科学研究科　〒 565–0871 大阪府吹田市山田丘 1-5

E-mail: †{wada.koji,harada.kei,sasaki,onizuka}@ist.osaka-u.ac.jp

あらまし　近年，GPS が自動車や携帯電話など様々なものに搭載され，位置情報を簡単かつ大量に取得することができる．それに伴い，オブジェクトの移動軌跡に関する研究も盛んに行われており，移動軌跡の中から周期的なパターンを検出する技術もその一つである．周期的なパターンの検出に関する既存研究はいくつか存在するが，どれも単一オブジェクトを対象としたもののみであり，複数オブジェクトの移動軌跡を考慮したものは存在しない．本稿では，複数オブジェクトの移動軌跡からの周期的なパターンの検出を目的とする．これに対し単純な手法を適用した場合，計算量が大きくなる．そのため，複数オブジェクトを対象とした効率的なアルゴリズムを提案する．提案アルゴリズムでは探索すべきパターンや空間を選別し，枝刈りを用いて計算量を減少する．実験には実世界のデータを用い、提案アルゴリズムが効率的に周期的なパターンを検出可能であることを示す．

キーワード　移動軌跡，時空間データマイニング，パターンマイニング

1. はじめに

近年，GPS センサが自動車やスマートフォンなどに搭載されるようになり，現在では様々なオブジェクトの位置情報が大量に収集可能である．そのため，位置情報を時系列順に並べた移動軌跡に関するデータマイニングの研究が盛んに行われている [1]．移動軌跡に関するデータマイニングにおいて，移動軌跡の周期的なパターン検出がある．例えば，ある人の行動を GPS データとして長期間収集した場合，その人が毎朝同じルートを通って会社に出勤する通勤ルートが周期的なパターンとして検出される．図 1 に周期的なパターンを示す．それぞれの点は，7 時から 7 時半の間のユーザの位置情報を表し，点が同じグリッド内にあるときそれらは同じ位置に存在するとする．周期を一日間とした場合，三日間に重複する移動軌跡があるため，それを周期的なパターンとして検出できる．

既存研究は単一のオブジェクトを対象とした周期的なパターンの検出を目的としている．人や動物の生活には習慣や規則性があると考えられるため [2]，対象とするものの性質を知る上で周期性を考慮したパターンマイニングは重要である．しかし，現在のところ既存研究では単一のオブジェクトを対象とした研究しか存在しない．図 2 は複数オブジェクトの移動軌跡を示し，同じ色であれば同じオブジェクトの移動軌跡である．このとき，単一のオブジェクトのみでは周期的なパターンは存在しないが，複数オブジェクトを考えた場合周期的なパターンが存在する．応用としては，複数の自動車の移動軌跡が収集できるとそれらのデータから特定の時期や時間によく利用される道路を発見し，交通状況の改善に役立てられる．複数のオブジェクトを対象とした周期的なパターンは，集団で存在するものの性質を把握する上で単一オブジェクトのみを対象としたパターン以上に重要であると考える．

本稿では，複数オブジェクトを対象とした周期的なパターンを定義する．また，定義したパターンを検出する際の課題を述べ，その課題を解決すべく効率的な探索手法を提案する．

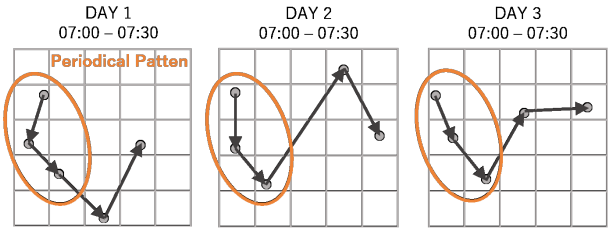

図 1　単一オブジェクトの移動軌跡

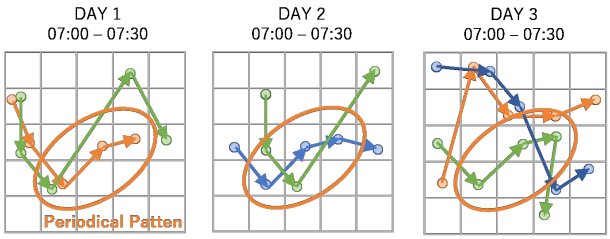

図 2　複数オブジェクトの移動軌跡

1.1 問題定義と技術概要

本稿では，複数オブジェクトの移動軌跡からの周期的なパターンの検出を目的とする．しかし，単一オブジェクトの移動軌跡とは異なる点がいくつか存在する．

- 同時刻に複数の移動軌跡が存在し得る．
- 同時刻に周期的なパターンが複数存在し得る．

既存研究が対象としている単一オブジェクトの移動軌跡のデータが，同時刻にただひとつの軌跡を持つことに対し，本研究で対象とする複数オブジェクトのデータは，同時刻に複数の移動軌跡が存在し得る．このことはパターン探索時の探索先のデータ量をオブジェクトの数に比例するように増加させ，計算時間を大きくする．さらに，同時刻に複数の移動軌跡が存在することにより，単一オブジェクトのみを考えたときと異なり同

時刻に周期的なパターンが複数存在し得る．既存研究が対象としている単一オブジェクトの移動軌跡の周期的パターンでは，1つの周期につき見つけるべきパターンは同時刻に 1 つだったため，パターンを見つけ次第探索を切り上げ，計算時間を抑えることができた．しかし，本稿で扱う複数オブジェクトの移動軌跡の場合，周期的なパターンとして同時刻に複数の移動軌跡が検出される可能性があり，探索中に周期的なパターンが検出できたとしてもその段階では探索を打ち切ることができず，再度他のパターンが存在するかどうかを探索する必要がある．よって，単一オブジェクトを扱う場合とは異なる方法で計算時間を削減しなければ，入力データ量の増加に対する計算時間の増加を抑えることができない．

本稿では 2 つの手法を提案する．探索先のデータ量が大きくなる問題に対しては，探索先のデータの集合が特定の地域情報を含むかどうかによって，探索すべきかどうかを判断し枝刈りを行う手法を提案する．探索回数が大きくなる問題に対しては，ボトムアップ的な手法を含む手法を用い，パターンとなり得ないようなものの探索を削減する手法を提案する．これら 2 つの手法は，検出すべきパターンが複数存在していても適用可能である．

1.2 本研究の貢献と構成

本研究の貢献を下記にまとめる．

- 複数のオブジェクトの移動軌跡の集合から周期的なパターンを検出する問題を新たに定義する．
- 複数のオブジェクトの移動軌跡からの周期的なパターンを効率的に検出可能なアルゴリズムを提案する．
- 実データを用いた実験より，提案アルゴリズムが周期的パターンを効率的に検出可能であることを示す．

本稿の構成は，次の通りである． 2. 章にて関連研究について述べる．3. 章で本稿の前提となる知識について概説する．4. 章にて提案手法の詳細について説明し，5. 章において提案手法の評価と分析を行う．6. 章にて本稿をまとめ，今後の課題について論ずる．

2. 関連研究

この章では，本稿の研究テーマである周期的なパターンマイニングとその他のデータマイニングに関して述べる．

移動軌跡を活用した研究は，行動予想 [3]，異常値検知 [4–6] や移動軌跡の分類 [7,8] など多岐にわたる．パターンマイニングも多数の研究が行われており，主に 4 つの主要なカテゴリが存在する [9]．ひとつは共に行動する対象を発見する Moving Together Pattern，次に似たような軌跡をまとめる Trajectory Clustering，3 つめに訪れた場所の順番に関する Sequential Pattern Mining があり，最後に本稿のテーマである周期的なパターンを発見する Periodical Pattern Mining が存在する．

移動軌跡に関する周期的なパターンマイニングには周期を指定する手法と自動で周期を特定する手法の 2 つの既存手法が存在し，大きく Fixed Period Approach と Reference Spot Approach の 2 つに分けられる [1].

Cao ら [10, 11] は任意の周期をあらかじめ与えることにより，周期的なパターンを発見する手法を提案している．位置情報の順列である 1 つの長い移動軌跡が与えられたとき，初めに指定した周期の長さで分割しセグメント化する．そして，セグメントにあらわれる閾値以上に出現するパターンを周期的なパターンとして検出する．その際まず長さが 1 のパターンを 1-patterns として検出し，ボトムアップ的にもしくはトップダウン的に探索し，最終的な周期的パターンを検出する．

Li ら [12,13] はユーザが周期を指定することなく，長い移動軌跡のデータから自動的に周期を特定する手法を提案している．まず，移動軌跡の座標情報の密度に対してカーネル法を用いてよく訪れているポイント（Reference Spot）を抽出し，それ以外の座標データと区別する．そして座標の順列で表されていた移動軌跡が， Reference Spot であるかそれ以外の場所であるかの 2 値で表現する．次に，変換された移動軌跡に対してフーリエ変換と自己相関を用いて周期を特定する．最後に，特定されたそれぞれの周期についてクラスタリングを用いて周期的なパターンを発見する．

これら 2 つの既存手法において，対象とされる移動軌跡は単一のオブジェクトが作る軌跡のみであり，複数のオブジェクトから生成される移動軌跡を包括的に取り扱い周期的なパターンを求めることは考えられていない．したがって，複数オブジェクトから生成される移動軌跡の周期的パターンを新たに考えたとき，単一オブジェクトでの手法がそのまま使えない場合や，計算量が大きくなってしまう問題がある．

また，実データの移動軌跡を扱う上で議論すべきの要素として位置情報の地域化がある．これは，実際の GPS データを扱う上で，座標データのあらゆる原因における数値の揺らぎを吸収するために行われる．Cao らは DBSCAN [14] のように密度ベースのクラスタリングを用いてグリッドベースの手法と比較実験を行っている．自動で周期を特定する手法においては，座標データの密度に対してカーネル法を用いて Reference Spot とそれ以外の地域に分けている．後者は周期を特定するための前処理でもあり，両者の地域化の意味合いは微妙に異なる．本稿では，複数オブジェクトの移動軌跡から周期的パターンを発見するために，グリッドベースの手法で地域化を行う．

3. 移動軌跡の周期的なパターン

この章では発見すべき移動軌跡の周期的なパターンについて定義する．また，単一オブジェクトの移動軌跡と複数オブジェクトの移動軌跡との違いについて，複数オブジェクトから発見される 2 種類の周期的なパターンについても言及する．

まず，位置の情報を座標情報として定義し，それらが順列に並んだ移動軌跡を定義する．そして，位置に関する情報の実際的なユレを吸収するための地域化について触れ，最終的に検出したい複数オブジェクトの周期的なパターンを定義する．

定義 1 (座標情報) 位置的な情報を持つ座標情報 l は $\{$ 緯度, 経度, タイムスタンプ$t\}$ の 3 つの要素で構成されている．また，タイムスタンプ t は 1 以上の整数とする．

定義 2 (移動軌跡) 移動軌跡 s は $s = \langle l_1, l_2, \ldots, l_n \rangle$ である．ただし，$l_{i+1}.t - l_i.t = 1$，$n \geqq 2$ である．

定義 3 (部分軌跡) ある移動軌跡 s を $\langle l_1, \ldots, l_n \rangle$ としたとき，$\langle l_i, l_{i+1}, \ldots, l_{j-1}, l_j \rangle (i \geqq 1, j \leqq n)$ を s の部分軌跡 s' とする．また，s 自身も s' である．ただし，部分軌跡は移動軌跡の条件を満たす．

ここで $S = \{s_1, s_2, \ldots, s_N\}$ と移動軌跡の集合が与えられたとき，S に含まれる全ての s の部分軌跡の集合を S' とする．ここからは，最終的なパターンの定義と取り扱う実際的なデータの性質に応じて必要となる定義をする．

座標情報 l は連続値であるため，特定の基準で離散的な値に区分することが必要となる．

定義 4 (地域) 空間的範囲を持つ位置的な情報 w を地域と呼ぶ．また，ある座標情報 l の緯度と経度が w の空間的範囲内に存在すれば，$\{w, t\}$ を地域化された座標 r と呼ぶ．このとき，l は r に属する．そして，移動軌跡の座標情報を地域化された座標に置き換えたものを，地域化された軌跡 s^r とする．また移動軌跡同様，$\langle r_i, r_{i+1}, \ldots, r_{j-1}, r_j \rangle (i \geqq 1, j \leqq n)$ を s^r の部分軌跡 $s^{r\prime}$ とする．

定義 5 (地域化された軌跡に属する移動軌跡) 移動軌跡を $s = \langle l_1, l_2, \ldots, l_n \rangle$ としたとき，$1 \leqq i \leqq n$ において，l_i が r_i に属していれば，移動軌跡 s は地域化された軌跡 $s^r = \langle r_1, r_2, \ldots, r_n \rangle$

複数オブジェクトの周期的なパターンを定義する前に，記号の定義とその説明をする．

S が与えられたとき周期 T を 1 以上の整数とし，m を 1 以上の整数とするとき，$L_S = m \cdot T$ は S の時間的長さであり，$L_S \geqq t$，$\forall l \in S$ かつ $(m-1) \cdot T \leqq t$ となるタイムスタンプを持つ l が S 内に存在する．例えば，1 日を 600 のタイムスタンプに分けたとき，$L_S = 6000$ であれば，10 日分の移動軌跡が S には含まれていることが分かる．

$C_i (0 \leqq i \leqq \frac{L_S}{T} - 1)$ は周期毎の移動軌跡の集合とする．このとき，$s \in C_i$ の長さは T 以下であり，s の座標情報のタイムスタンプは $i \cdot T < t \leqq (i+1) \cdot T$ を満たす．

定義 6 (移動軌跡の周期的なパターン) 移動軌跡の集合 S，その L_S，周期 T，周期毎の移動軌跡の集合 $C_i (0 \leqq i \leqq \frac{L_D}{T} - 1)$，閾値 θ_{th}，θ_{low} が与えられたとする．ただし，$C_0 \cup C_1 \cup \ldots \cup C_{\frac{L_S}{T}-1} = S$ である．また，$p = \langle r_1, r_2, \ldots, r_m \rangle (r_m.t \leqq T)$ としたとき，p のそれぞれの r のタイムスタンプ t に $i \cdot T$ を加えたものを $q_i (0 \leqq i \leqq \frac{L_D}{T} - 1)$ とする．このとき，q_i に属する s' が C_i' 内に θ_{low} 本以上存在するという条件を満たす C_i の数が $\frac{L_D}{T} \cdot \theta_\alpha$ 以上であれば，p を移動軌跡の周期的なパターンとする．

定義 7 (極大パターン) 周期的パターン p において，p にいかなる r を加えても周期的パターンとならない場合，p を極大パターンとする．

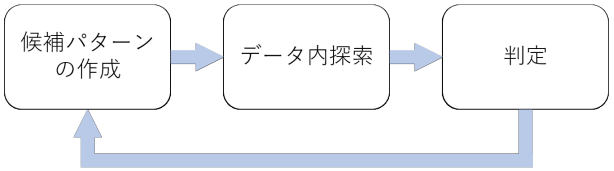

図 3　検出手順の概略

定義 8 (周期的なパターンマイニング) 移動軌跡の集合 S，周期 T，閾値 θ_{th}，θ_{low} が与えられたとき，全ての周期的な極大パターンを検出する．

4. 提案手法

本稿では，複数オブジェクトから生成された移動軌跡 s の集合 S が入力データとして与えられたとき，新たに定義した周期的なパターンを発見する．

4.1 節にて検出手順を示し，複数オブジェクトの移動軌跡を入力とした特徴や問題点について論じる．また，入力データである移動軌跡の集合 S と本稿における座標の地域化についても論じる．4.3 節，4.4 節では 4.1 節で述べた問題点や特徴に対応した手法を提案する．そして 4.5 節ではアルゴリズムについて説明する．

4.1 検出手順

周期的パターン検出手順の概略を図 3 に示す．初めに候補パターンを作成し，次にその候補パターンと同じ軌跡がデータ内に存在するか探索する．あらかじめ決めた閾値以上に存在すれば周期的パターンであると認定し，存在しなければ周期的パターンとはしない．これを有効な候補パターンが作成不可になるまで繰り返すことで，全ての極大パターン検出を目指す．

4.1.1 前処理

各ステップ毎に詳しく述べる前に，検出や提案手法の実行のための前処理について述べる．

入力データとして移動軌跡の集合 S が与えられたとき，それを $C_i (0 \leqq i \leqq \frac{L_S}{T})$ に分割する．これにより，任意の周期 T 毎に移動軌跡を管理するのだが，S の中には周期をまたぐ移動軌跡も存在する．その場合は周期の境目でその移動軌跡が分割され，それぞれの C_i に振り分けられる．よって，前処理された後の移動軌跡の長さは全て C_i の時間的長さ以下となる．10 日分のデータを入力とし，周期を 1 日と設定したとき，入力データは図 4 のように日ごとに管理される．

次に，座標と移動軌跡の地域化について述べる．座標データは連続値であり，それらの一致をいうためには一定の基準を設けなければならない．今回は地域という離散な値に変換することでその基準を設ける．今回提案する手法は地域の分け方によらず適用可能であるため，単純なグリッドベースにより地域化する．なお，実装時はグリッドの範囲，水平方向と垂直方向のグリッドの数を定め，それぞれに等分している．そのため，各座標につき 1 つのグリッドが対応し，前処理において座標をグリッドの値に変換している．

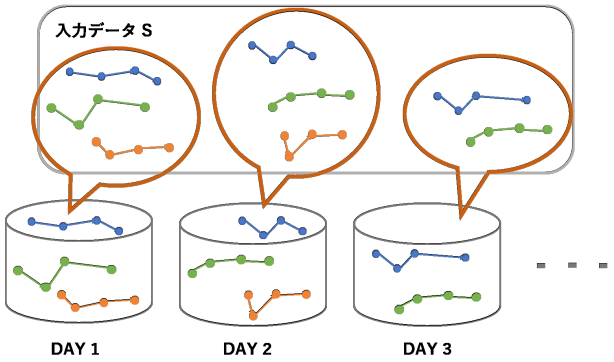

図 4　移動軌跡を分割して管理

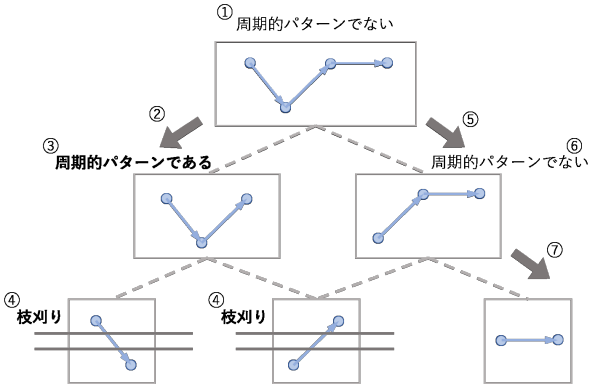

図 5　トップダウンによる枝刈り

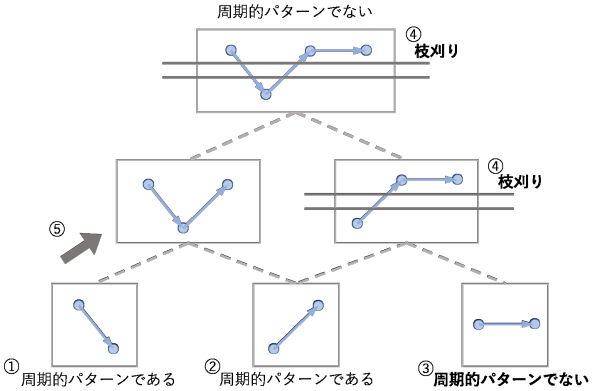

図 6　ボトムアップによる枝刈り

4.1.2　候補パターンの作成

候補パターンをどのように作成し，どのようなパターンを作成するかは計算時間削減に大きく影響する．単純に考えると，S 内の全ての移動軌跡 s の部分軌跡集合 S' を全て候補パターンとすればよい．しかしその場合，移動軌跡の長さ，入力 S の大きさが増大することで候補パターンの数が爆発的に増加する．候補パターンの増加は検出手順のループ回数に結びつき，全体的な実行時間が著しく増大することになる．一方，候補パターンの数を減らす方法を考える．例えば，S' の中からランダムに複数の軌跡を選出し，それらを候補パターンから外すことを考える．その場合，S' の大きさが減る，つまり候補パターンの数が減るため，検出手順のループ回数も減少し，全体の実行時間が短縮される．しかし，本来であれば周期的パターンとして検出されるはずであった候補も外す可能性があるため，不用意に候補パターンを削減することはできない．

4.1.3　データ内探索

データ量の増加は探索する空間の拡大に直接結びつくため，大規模なデータを扱う上で有効な探索法を提案することは重要である．本稿では，4.1.1 項で述べたように周期毎に移動軌跡を管理している．よって単純な手順として，C_0 から $C_{\frac{L_S}{T}}$ まで順にすべて探索し，閾値以上であれば周期的パターンとする方法が考えられる．既存研究で扱われている，単一なオブジェクトを対象としたものであっても周期毎に分割されるが，複数オブジェクトの移動軌跡を入力として許すと単一オブジェクトとは違い，C_i が保持する移動軌跡の数が定まらない．S の増大に伴ってその数が増加し，単純な手法だと探索にかかる時間も大きくなると考えられる．

4.2　ベースライン

既存研究 [10] では，トップダウン的に候補パターンを作成する．本稿ではこれをベースラインの候補パターン作成手法とする．また，データ内探索ステップでは単純に C_i を全て探索する手法をベースラインに用いる．トップダウンで候補パターンを作成すると，p が周期的パターンであった場合その部分軌跡である p' 以降の候補パターン作成を省くことができる．トップダウンによる候補パターン作成と枝刈りの様子を図 5 に示す．図中の丸で囲まれた各数字はトップダウンで候補パターンを作成する際の手順を表す．長さの長い軌跡から候補パターンとし，そのパターンが周期的に現れているかをデータ内を探索して確認する．周期的パターンであった場合，極大パターンの定義により，それより短い軌跡については候補パターンとせず，候補パターン作成を枝刈りする．

4.3　ボトムアップ

候補パターン作成時の手法を提案する．複数のオブジェクトにおいて周期的パターンを検出する際は，単一の場合と異なりひとつひとつのオブジェクトが周期的でない移動軌跡を多く含む可能性がある．また，周期的パターンではない $s^{r'}$ を含む軌跡 s^r は周期的パターンとなり得ない．よって，無駄な候補パターン作成を抑制する手法として，短い軌跡から候補パターンを作成するボトムアップが効果的だと考える．図 6 にボトムアップによる候補パターンの作成と枝刈りの様子を示す．図中の丸で囲まれた各数字はボトムアップで候補パターンを作成する際の手順を表す．まず長さが一番短い軌跡を候補パターンとし，それが周期的に出現するかをデータ内を探索して調べる．調べ終えると次にそれらを結合し，長さが 1 長い軌跡を候補パターンとしていくのだが，先ほど述べたように周期的パターンでない軌跡は周期的パターンになり得ないので，候補パターン作成に関して枝刈りが可能である．よって，候補パターン集合の大きさを小さくすることが可能である．候補パターンの数が減ると，検出手法全体のループ回数が減少し実行時間を抑えられる．図 6 にボトムアップによる候補パターンの作成と枝刈りの様子を示す．

4.4　空間的枝刈り

先の 4.3 節では候補パターンを作成する段階においての問

題に対処する手法を提案したが，ここではデータ内を探索する段階に生じるであろう問題に対処する手法を提案する．データ内を探索する際は，周期毎に分けられた移動軌跡の集合を順に調べていくのだが，オブジェクトの数に比例して各集合に含まれる移送軌跡が増加し，探索にかかる時間も大きくなる問題があった．その問題に対して，本稿では各集合 C_i にそれぞれの集合に含まれる地域の情報を持たせ，その情報を用いて C_i が探索すべき集合かどうかを判断し，枝刈りを行う手法を提案する．

この手法を行うため，4.1.1 項で説明した前処理時に各 C_i には自身が含む全移動軌跡が持つ地域をメタ情報として付与する．そして，データ内探索時に探索しようとしている候補パターンが持つ地域が探索先の C_i に含まれているかによって，C_i を探索するかしないかを決め枝刈りする．

4.5 アルゴリズム

以上で提案手法の概略を述べた．この節では周期的パターン検出の全体的なアルゴリズム，また各手法のアルゴリズムについて説明する．

アルゴリズム 1 は各手法に共通する前処理を含む，フレームワークの手順である．1 行目の convert は 4.1.1 項で説明したように，地域集合 R に各座標を地域化し，移動軌跡の分割を行う関数である．2 行目から 13 行目で $s^r \in S^r$ を周期毎の集合 C_i に分割する．最後に，15 行目から 22 行目で候補パターンの作成，データ内探索を行い周期的なパターンを検出する．候補パターンは bottomup，topdown でボトムアップ，トップダウンの各手法に沿って作成する．

アルゴリズム 2 と 3 は候補パターン作成時の各手法のアルゴリズムである．search とはデータ内探索を行う関数であり，アルゴリズム 4 と 5 に提案手法である空間的枝刈りの内容を示す．アルゴリズム 2 は，ベースラインとするトップダウンのアルゴリズムである．3 行目から 4 行目では，$s^{r\prime}$ についてデータ内探索を行い，周期的パターンであればそのパターンを返す．5 行目から 11 行目までは $s^{r\prime}$ が周期的パターンでない場合の処理である．9 行目で $s^{r\prime}$ より長さが 1 短い部分軌跡を $s^{r\prime}$ から作成する．そして 10 行目で，先ほど作成した $s^{r\prime}$ の集合 $S^{r\prime}$ を入力として再帰的に topdown を呼び出す．アルゴリズム 3 は提案手法であるボトムアップのアルゴリズムである．2 行目から 6 行目までは，$s^{r\prime} \in S^{r\prime}$ が周期的パターンであるか判定するため，seartch でデータ内を探索する．周期的パターンが検出されなかった場合，ここで候補パターンの作成を終える．7 行目から 13 行目までは，2 行目から 6 行目までにおいて周期的なパターンが検出できた場合の処理である．11 行目では，検出された周期的なパターンの中で共通のタイムスタンプ t を持つパターンを結合し，新たに長い候補パターンを作成する．そして新たに作成した候補パターンの集合 $S^{r\prime}$ を入力とし，再帰的に bottomup を呼び出す．

アルゴリズム 4 と 5 は空間的枝刈りに関連するアルゴリズムである．アルゴリズム 4 はデータ内を探索するアルゴリズムである．1 行目で valid_set を呼び出し，探索すべき集合を取得する．3 行目から 8 行目で，探索すべき集合 $C_i \in V$ を探索先として，候補パターンに一致する軌跡が含まれているか探索する．そして，探索先に一致する軌跡が θ_{low} 以上存在する C_i の数が $|\mathbf{C}| \cdot \theta_{th}$ 以上であれば $s^{r\prime}$ を周期的パターンとする．アルゴリズム 5 は探索すべき集合を判定するアルゴリズムである．2 行目から 6 行目で，条件を満たす C_i を有効な探索すべき集合とする．そして 7 行目から 9 行目で，有効な集合の数 $|V|$ が $|\mathbf{C}| \cdot \theta_{th}$ 以上であれば，V 内の集合を探索すべき集合として返す．

Algorithm 1 Periodical Pattern Mining$(S, T, R, \theta_{th}, \theta_{low})$

```
1: S^r ← convert(S, R, T)
2: for i = 0 to L_D/T do
3:     C_i = φ
4:     C ← C_i
5: end for
6: for all s^r ∈ S^r do
7:     for i = 0 to L_D/T do
8:         if the timestamps of s^r ∈ [i·T, (i+1)·T − 1] then
9:             C_i ← C_i ∪ s^r
10:            break
11:        end if
12:    end for
13: end for
14: for all s^r ∈ S^r do
15:    if bottomup then
16:        S^r' ← ∀l ∈ s^r
17:        P ← bottomup(s^r, S^r', C, θ_th, θ_low)
18:    else topdown
19:        S^r' ← s^r
20:        P ← topdown(s^r, S^r', C, θ_th, θ_low)
21:    end if
22: end for
```

Algorithm 2 topdown$(s^r, S^{r\prime}, \mathbf{C}, \theta_{th}, \theta_{low})$

```
1: P ← φ
2: for all s^r' ∈ S^r' do
3:     if search(s^r', C, θ_th, θ_low) then
4:         return s^r'
5:     else
6:         if |s^r'| is 1 then
7:             return
8:         end if
9:         S^r' ← {s^r' − l_0, s^r' − l_|s^r'|}
10:        P ← P∪ {topdown(s^r, S^r', C)}
11:    end if
12: end for
13: return P
```

5. 評価実験

本章では提案手法の評価として効率性を評価する．実験は Intel Core i7-8700K 3.70 GHz の CPU，メモリは 64 GB のものを載せたコンピュータ上で行う．また，全ての手法は C++ で実装した．移動軌跡のデータセットとして，北京市内のタクシーから得られた GPS の軌跡データ T-drive [15, 16] を用い

Algorithm 3 bottomup($s^r, S^{r\prime}, \mathbf{C}, \theta_{th}, \theta_{low}$)

```
P ← φ
for all s^r' ∈ S^r' do
    if search(s^r', C, θ_th, θ_low) then
        P ← P ∪ s^r'
    end if
end for
if P ≠ φ then
    if P includes s^r then
        return P
    end if
    S^r' ← join any pairs of periodical patterns
    P ← P∪ {bottomup(s^r, S^r', C)}
end if
return P
```

Algorithm 4 search($s^{r\prime}, \mathbf{C}, \theta_{th}, \theta_{low}$)

```
V ← valid_set(s^r', C, θ_th, θ_low)
count = 0
for all C_i ∈ V do
    m = |{s^r' | s^r' ∈ s^r ∈ C_i}|
    if m ≧ θ_low then
        count = count + 1
    end if
end for
if count ≧ |C| · θ_th then
    return true
end if
return false
```

Algorithm 5 valid_set($s^{r\prime}, \mathbf{C}, \theta_{th}, \theta_{low}$)

```
V ← φ
for C_i do
    if C_i include all regions that s^r' include then
        V ← V ∪ C_i
    end if
end for
if |V| ≧ |C| · θ_th then
    return V
end if
return φ
```

る．このデータセットには 2008 年 2 月 2 日から 8 日まで北京市内を走行した 10,357 台のタクシーのデータが含まれており，タクシーの識別子，緯度経度とタイムスタンプから構成される．実験ではこのデータを 1 日を 100 としたタイムスタンプに分けサンプリングしたものを使用する．前処理における地域化の方法はグリッドベースを用い，実験を通して固定された枠内を水平，垂直方向それぞれ g_h，g_v 個に等間隔で分割する．

評価実験を通して，特に記述がない場合はデフォルトのパラメータとして，$T = 100$，$\theta_{th} = 0.5$，$theta_{low} = 1$，$g_h = 3000$，$g_v = 3000$ を使用する．また，2 つの提案手法を組み合わせた複合手法についても検討する．

5.1 効 率 性

まず，各提案手法がベースラインと比較し高速に周期的なパターンを検出できることを示す．図 7 はデータセットより 1000 本の移動軌跡を作成し，検出対象として入力したときの各手法による実行時間の比較である．これによりベースラインと比べ，各提案手法は数倍から数十倍高速にパターンを検出していることが分かる．

候補パターンの作成時の違いであるベースラインとボトムアップを比較する．ボトムアップの方がベースラインに比べて効率的にパターンを検出できている．そのため，まず短い部分軌跡を調べそれが周期的なパターンでなかったときの候補パターンの枝刈りが効果的に行われたと言える．次にデータ内探索時の違いである，ベースラインと空間的枝刈り手法とを比較する．実行時間が減少していることから，効果的に探索すべき集合を選択できていると考えられる．最後に提案手法両方を組み合わせた複合手法について述べる．実行時間から，これが最も効率的に検出を行えていると言えるが，ベースラインと空間的枝刈り手法との差ほどボトムアップとの差がない．この要因として空間的枝刈り手法の特徴が挙げられる．提案した空間的枝刈り手法は，候補パターンが持つ地域情報に基づいて枝刈りを行う手法であり，長い候補パターンが様々な地域を含んでいるとそれだけ探索先の集合に課す条件が厳しくなる．例えば候補パターンが $r_1 = \langle\{w_1, 1\}, \{w_2, 1\}, \{w_3, 3\}, \{w_4, 4\}\rangle$ としたとき，探索先の集合 C_i は w_1，w_2，w_3，w_4 全ての地域を含んでいる必要があり，候補パターンを $r_2 = \langle\{w_1, 1\}, \{w_2, 1\}\rangle$ としたときに比べ，探索先が満たす条件は厳しくなる．また，ボトムアップは効率よく候補パターンを削減できているため，その手法が作成した候補パターンは短い段階で枝刈りされていると考えられる．よって，ボトムアップに空間的枝刈りを併用すると短い候補パターンをもとに探索先を枝刈りすることが比較的多くなる．よって緩い条件で空間的枝刈りを試みる比率が，トップダウンで候補パターンを作成するときより大きくなったと考えられる．そのためトップダウンに空間的枝刈りを行ったほどの効果がなかったのだろう．

次に，入力データの大きさを変化させたとき，各手法の実行時間を計測する．図 8 は入力データ S に含まれる移動軌跡の本数を 1000 から 10000 まで変化させた場合の実行時間である．ベースラインでは入力データが大きくなると，実行時間が大幅に増加していることが分かる．各提案手法ではベースラインに比べ実行時間自体は抑えられているが，入力データの増加に伴い実行時間が大きく増加する傾向がある．提案手法はベースラインに比べ，実行時間を定数倍減らすことは可能だが，より巨大なデータに対して直接適用できない可能性がある．

実験を通して，最も高速であるのは複合手法である．これは，ボトムアップで候補パターンを作成し，その後の探索時に空間的枝刈りを行うものである．よって，探索時の手法の違いのみであるベースラインと空間的枝刈り手法との間の差と同じような差があってよいはずである．しかし，その差は数倍程度に収まっている．実験結果から，ボトムアップは候補パターンの数を，空間的枝刈りは探索する集合の数を減らすことができたと

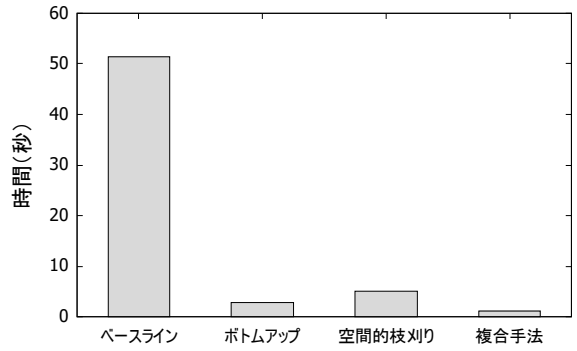

図 7　移動軌跡 1000 本を入力としたときの実行時間

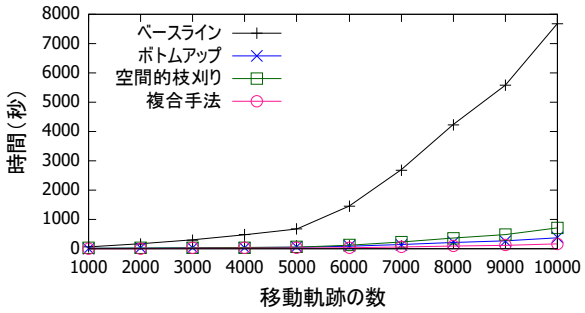

図 8　移動軌跡の数を変化させたときの実行時間

言える．

6. ま と め

本稿では複数オブジェクトから生成された移動軌跡の周期的なパターンを新たに定義した．さらにその周期的なパターンを検出することを目的とし，実用的な計算時間で処理を終えるためボトムアップと空間的枝刈りの 2 つの手法を提案した．評価実験ではベースラインとした手法と比較し，提案手法が計算時間を削減し効率的に周期的なパターンを検出可能であることを示した．

今後の研究として三つの課題に取り組む．まず，計算量の削減が十分ではないため，大規模なデータセットに適応することがまだ難しい．そのため，事前計算を用いることや並列分散計算などを用いたさらなる効率化が必要である．次に，地域の分割方法について十分な考察ができていない．本稿では座標情報を扱う際にグリッドベースで地域に分割して位置情報を扱ったが，その他の分割法を用いることにより，新たに有用なパターンが検出できる可能性がある．そのため地域で位置情報を扱う方法について，より考察する必要がある．最後に，使用するデータの傾向を考える必要がある．例えば，都市を移動するタクシーと鳥の群れでは移動傾向が全く異なる．そのため，データの性質を考慮した手法を考案する必要がある．

謝　　辞

本研究は科学研究費（JP16H01722）の支援によって行われた．ここに記して謝意を表す．

文　　献

[1] Dongzhi Zhang, Kyungmi Lee, and Ickjai Lee. Periodic pattern mining for spatio-temporal trajectories: a survey. In *Proceedings of Intelligent Systems and Knowledge Engineering (ISKE)*, pp. 306–313, 2015.

[2] Xiaopeng Chen, Dianxi Shi, Banghui Zhao, and Fan Liu. Periodic pattern mining based on gps trajectories. In *Proceedings of International Symposium on Advances in Electrical, Electronics and Computer Engineering*, pp. 181–187, 2016.

[3] Anna Monreale, Fabio Pinelli, Roberto Trasarti, and Fosca Giannotti. Wherenext: a location predictor on trajectory pattern mining. In *Proceedings of ACM SIGKDD*, pp. 637–646, 2009.

[4] Lei Zhang, Zimu Hu, and Guang Yang. Trajectory outlier detection based on multi-factors. *IEICE Transactions on Information and Systems*, Vol. 97, No. 8, pp. 2170–2173, 2014.

[5] Prithu Banerjee, Pranali Yawalkar, and Sayan Ranu. Mantra: a scalable approach to mining temporally anomalous sub-trajectories. In *Proceedings of ACM SIGKDD*, pp. 1415–1424, 2016.

[6] Yanwei Yu, Lei Cao, Elke A Rundensteiner, and Qin Wang. Detecting moving object outliers in massive-scale trajectory streams. In *Proceedings of ACM SIGKDD*, pp. 422–431, 2014.

[7] Yin Zhu, Yu Zheng, Liuhang Zhang, Darshan Santani, Xing Xie, and Qiang Yang. Inferring taxi status using gps trajectories. *arXiv preprint arXiv:1205.4378*, 2012.

[8] Timothy Sohn, Alex Varshavsky, Anthony LaMarca, Mike Y Chen, Tanzeem Choudhury, Ian Smith, Sunny Consolvo, Jeffrey Hightower, William G Griswold, and Eyal De Lara. Mobility detection using everyday gsm traces. In *Proceedigns of International Conference on Ubiquitous Computing*, pp. 212–224, 2006.

[9] Yu Zheng. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology*, Vol. 6, No. 3, p. 29, 2015.

[10] Huiping Cao, Nikos Mamoulis, and David W Cheung. Discovery of periodic patterns in spatiotemporal sequences. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 19, No. 4, pp. 453–467, 2007.

[11] Nikos Mamoulis, Huiping Cao, George Kollios, Marios Hadjieleftheriou, Yufei Tao, and David W Cheung. Mining, indexing, and querying historical spatiotemporal data. In *Proceedings of ACM SIGKDD*, pp. 236–245, 2004.

[12] Zhenhui Li, Jiawei Han, Ming Ji, Lu-An Tang, Yintao Yu, Bolin Ding, Jae-Gil Lee, and Roland Kays. Movemine: Mining moving object data for discovery of animal movement patterns. *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 4, p. 37, 2011.

[13] Zhenhui Li, Jiawei Han, Bolin Ding, and Roland Kays. Mining periodic behaviors of object movements for animal and biological sustainability studies. *Data Mining and Knowledge Discovery*, Vol. 24, No. 2, pp. 355–386, 2012.

[14] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of ACM SIGKDD*, Vol. 96, pp. 226–231, 1996.

[15] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. T-drive: driving directions based on taxi trajectories. In *Proceedings of SIGSPATIAL*, pp. 99–108, 2010.

[16] Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. Driving with knowledge from the physical world. In *Proceedings of ACM SIGKDD*, pp. 316–324, 2011.